

Ricerca di Sistema elettrico



Interfacciamento del navigatore con il sistema di navigazione autonoma del veicolo elettrico e integrazione del modulo di riconoscimento della sede stradale (LA2.22)

Gabriele Costante, Simone Felicioni, Mario Luca Fravolini e Paolo Valigi

A.D. 1308

unipg

DIPARTIMENTO
DI INGEGNERIA

Interfacciamento del navigatore con il sistema di navigazione autonoma del veicolo elettrico e integrazione del modulo di riconoscimento della sede stradale

LA2.22 - Riconoscimento della sede stradale e interfacciamento con un navigatore

G. Costante, S. Felicioni, M.L. Fravolini, P. Valigi (Dipartimento di Ingegneria – Università degli Studi di Perugia)

Dicembre 2024

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell’Ambiente e della Sicurezza Energetica - ENEA Piano Triennale di Realizzazione 2022-2024

Obiettivo: Decarbonizzazione

Progetto: Tema di ricerca 1.7 – Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Linea di attività: 2.22

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work Package: Claudia Meloni, ENEA

Responsabile Linea di Attività: Paolo Valigi

Mese inizio previsto: Luglio 2023

Mese inizio effettivo: Luglio 2023

Mese fine previsto: Dicembre 2024

Mese fine effettivo: Dicembre 2024

Il presente documento descrive le attività di ricerca svolte all’interno dell’Accordo di collaborazione: “Affinamento del software di pianificazione e navigazione del veicolo, riconoscimento della sede stradale e interfacciamento con un navigatore”

1	Risultati attesi	4
2	Risultati ottenuti	5
2.1	Interfacciamento tra il navigatore (pianificatore globale) e il sistema di navigazione: 5	
2.2	Segmentazione dinamica del manto stradale e correzione delle traiettorie:	5
2.3	Sperimentazioni pratiche e validazione:	5
2.4	Integrazione finale e consegna:.....	5
3	Prodotti attesi	6
4	Prodotti sviluppati.....	7
5	Analisi degli scostamenti su attività e risultati	8
6	Sintesi delle attività svolte.....	9
7	Dettaglio delle attività svolte	10
7.1	Introduzione al sistema di navigazione finale	10
7.2	Modulo di Localizzazione	11
7.3	Modulo Pianificazione globale	12
7.4	Modulo di Pianificazione Locale.....	13
7.5	Modulo di Generazione della Mappa e Segmentazione del Manto Stradale.....	14
7.6	Controllore	17
7.7	Integrazione dei moduli e ROS.....	17
7.8	Setup Sperimentale	18
7.9	Risultati ottenuti	19
8	Contributo delle eventuali consulenze alle attività sopra descritte	21
9	Pubblicazioni scientifiche	22
10	Eventi di disseminazione.....	23

Figura 1: sistema di navigazione: il punto di arrivo viene selezionato tramite interfaccia grafica e, successivamente, il pianificatore globale GraphHopper genera una sequenza di waypoints fra la posizione attuale del veicolo ed il punto di arrivo. Il modulo di pianificazione locale calcola un percorso che congiunge gli waypoints restituiti dal pianificatore globale ed evita al contempo gli ostacoli. Il controllore in retroazione garantisce l'inseguimento del percorso fornito dal pianificatore locale. 10

Figura 2: stima della posa attraverso GPS-RTK (in rosso), DLO (in blu), e fusione sensoriale (in verde). 12

Figura 3: il modulo di conversione delle coordinate è una componente fondamentale del pianificatore globale in quanto, a partire dalla sequenza di waypoints in coordinate georeferenziate, si occupa di convertirle prima in un sistema di riferimento cartesiano attraverso la proiezione di Gauss-Krüger, per poi convertirle nel sistema di riferimento locale relativo alla posizione iniziale del veicolo..... 13

Figura 4: esempi di percorsi generati da RRT*-Dubins. L'algoritmo genera un albero di rami candidati (in verde), tra i quali seleziona il percorso con costo minore per arrivare a destinazione (in blu). 14

Figura 5: durante l'inseguimento del percorso generato dal pianificatore locale, questo modulo si occupa di rilevare possibili collisioni con ostacoli lungo il percorso stesso. In caso positivo, il veicolo viene fermato finché non viene ripianificato un nuovo percorso che eviti l'ostacolo identificato. 14

Figura 6: mappa 3D generata da DLO a partire da misure LiDAR. 15

Figura 7: esempio di segmentazione del manto stradale (in rosso) a partire dalle misure LiDAR. 16

Figura 8: il controllore di Stanley agisce congiuntamente l'errore di traiettoria (ψ) e l'errore di posizione (e) per l'inseguimento del percorso. 17

Figura 9: grafo delle connessioni dei nodi ROS nell'architettura proposta..... 18

Figura 10: veicolo autonomo utilizzato nella campagna sperimentale. 19

Figura 11: veicolo utilizzato durante una sperimentazione di guida autonoma. 19

Figura 12: esempi di navigazione autonoma all'interno del Centro Ricerche Casaccia. Il controllore si occupa di inviare i comandi ai motori per annullare l'errore di inseguimento tra il percorso desiderato (in blu) e la posa del veicolo (in verde). 20

1 Risultati attesi

Risultati della LA2.22: "Riconoscimento della sede stradale e interfacciamento con un navigatore"

Le attività della LA 2.22 si articolano in due filoni fondamentali. Nel primo filone, sulla base di quanto compiuto nella precedente LA, si aggiornerà il sistema di interfacciamento tra il navigatore precedentemente individuato e la pipeline di navigazione del veicolo. Saranno sviluppate metodologie per l'elaborazione di percorsi (codificati ad esempio tramite waypoints) a partire da coordinate GPS che descrivono il punto di arrivo desiderato (latitudine e longitudine), dall'altro l'elaborazione di traiettorie compatibili con il percorso e la conseguente navigazione. Allo stato attuale il veicolo possiede una rappresentazione del mondo circostante espressa in termini metrici e gli obiettivi di navigazione sono quindi forniti attraverso coordinate cartesiane. Lo scopo di questa attività sarà quindi quello di sviluppare un sistema in grado di localizzare il veicolo ed elaborare percorsi rispetto a coordinate georeferenziate (GPS - latitudine/longitudine). In particolare, si prevede che il sistema elabori waypoints forniti da sistemi di planning e li adatti alla propria pipeline di navigazione. Quest'ultima avrà come scopo il tracking della traiettoria calcolata a partire da input georeferenziate, anche attraverso tecniche di mappatura locale metrica, Simultaneous Localization and Mapping (basato ad esempio su Laser o sistemi di telecamere) e local re-planning. I sistemi di planning prenderanno in considerazione percorsi elaborati a partire da navigatori basati su mappe stradali (es. open street map) se disponibili per i luoghi previsti per la sperimentazione.

Nel secondo filone saranno studiate ed utilizzate tecniche per il riconoscimento della sede stradale ai fini di migliorare le performance del sistema di navigazione. In un primo momento verranno valutate le prestazioni del sistema sviluppato nelle fasi precedenti, misurando la sua robustezza rispetto al mantenimento del veicolo all'interno della carreggiata stradale. Le traiettorie elaborate a partire da input georeferenziate e le incertezze legate ai sistemi di localizzazione e navigazione del veicolo potrebbero infatti non garantire che quest'ultimo si mantenga sempre all'interno della carreggiata. Si procederà quindi all'implementazione di approcci di segmentazione del manto stradale a partire da immagini e/o altri dati forniti dai vari sensori. La segmentazione verrà quindi utilizzata per correggere le traiettorie del veicolo per evitare che questo, durante la navigazione, esca dalla carreggiata.

In parallelo alle attività è prevista la presenza di personale dell'Università presso il C.R. Casaccia per le sperimentazioni e la messa a punto dei parametri del software sul veicolo in scenari di percorsi semplici con e senza ostacoli.

I risultati previsti possono essere quindi riassunti come segue:

- Versione aggiornata del sistema di interfacciamento tra navigatore open source e pipeline di navigazione del veicolo.
- Sviluppo di metodologie per la pianificazione e navigazione su strada a partire da coordinate georeferenziate (waypoints espressi in termini di longitudine e latitudine).
- Sistema di segmentazione del manto stradale e sistema di correzione della traiettoria (local re-planning) sulla base dell'input fornito dal sistema di segmentazione.

2 Risultati ottenuti

2.1 Interfacciamento tra il navigatore (pianificatore globale) e il sistema di navigazione:

È stato sviluppato un sistema di navigazione avanzato che integra moduli di pianificazione globale con un'interfaccia grafica intuitiva (GUI) basata su Graphhopper. Questo sistema permette all'utente di selezionare in modo semplice i punti di destinazione per il veicolo autonomo. I waypoint forniti da Graphhopper vengono convertiti in coordinate cartesiane e processati dal modulo di pianificazione locale. Quest'ultimo elabora traiettorie compatibili con il modello cinematico del veicolo, garantendo una navigazione fluida verso i waypoint designati.

Le traiettorie calcolate vengono seguite mediante un controllore, che genera i comandi di basso livello necessari per il movimento del veicolo. Il controllore confronta costantemente la posizione stimata del veicolo, fornita dal sistema di localizzazione, con quella prevista dalla traiettoria, minimizzando gli errori.

2.2 Segmentazione dinamica del manto stradale e correzione delle traiettorie:

È stata implementata la segmentazione del manto stradale utilizzando dati lidar. Il sistema è in grado di distinguere dinamicamente tra ostacoli e aree percorribili, migliorando significativamente la sicurezza e l'efficacia della navigazione. I dati provenienti dalla segmentazione sono stati utilizzati per correggere in tempo reale le traiettorie del veicolo, attraverso un modulo di local re-planning, garantendo il mantenimento all'interno della carreggiata anche in presenza di incertezze o perturbazioni.

2.3 Sperimentazioni pratiche e validazione:

Sono state condotte sperimentazioni presso il Centro Ricerche Casaccia, con la partecipazione attiva del personale universitario. Durante queste prove il sistema è stato configurato, testato e ottimizzato in scenari di percorsi semplici, sia con che senza ostacoli. I test hanno dimostrato l'efficacia del sistema, che ha navigato autonomamente in base agli input forniti dall'interfaccia utente.

2.4 Integrazione finale e consegna:

Il software sviluppato è stato completamente integrato nella piattaforma veicolare e rilasciato contestualmente al presente report.

3 Prodotti attesi

I prodotti attesi relativi alle attività della LA2.22 sono i seguenti:

- Report di fine attività che conterrà la descrizione delle attività svolte nell'ambito della ricerca su "Riconoscimento della sede stradale e interfacciamento con un navigatore" ed i relativi risultati raggiunti.
- **Software**: codice sviluppato per la realizzazione dei PoC (Proof of Concept) relativi alle attività elencate precedentemente e alla loro validazione.

4 Prodotti sviluppati

- È stato realizzato il presente report delle attività in cui sono dettagliate le metodologie sviluppate e i dettagli del sistema di navigazione. In particolare, il report descrive i vari moduli del sistema di navigazione per il veicolo autonomo, inclusi i moduli di interfacciamento tra il pianificatore globale basato su GraphHopper che mette a disposizione dell'utente un'interfaccia di navigazione, e il modulo per il riconoscimento del manto stradale. Inoltre, il report descrive i risultati dei test effettuati al Centro Ricerche Casaccia, presso il quale i ricercatori dell'Università degli Studi di Perugia si sono recati in più occasioni sia per configurare i moduli software sia mettere a punto i parametri dei moduli di navigazione ed effettuare le sperimentazioni con il veicolo autonomo.
- È stato inoltre completato il software relativo al sistema di navigazione. Il software comprende tutti i moduli descritti nel presente report e i relativi nodi ROS. Il software, oltre ad essere stato integrato nel veicolo del Centro Ricerche Casaccia, è anche fornito in allegato al presente report come deliverable della LA2.22.

5 Analisi degli scostamenti su attività e risultati

Non si riscontrano scostamenti rilevanti tra le attività pianificate e quelle effettivamente svolte.

6 Sintesi delle attività svolte

Nella LA2.22 è stato completato e integrato nel veicolo il sistema di navigazione autonoma. Le attività si sono articolate in due filoni principali: il primo ha riguardato la realizzazione e l'implementazione del sistema di navigazione. I moduli di pianificazione globale e locale sono stati integrati con l'interfaccia grafica basata su Graphhopper sviluppata nella LA2.21. I waypoints generati vengono quindi elaborati per ottenere traiettorie compatibili con la dinamica del veicolo. Un controllore garantisce il tracking delle traiettorie, minimizzando errori di posizione e orientamento, stimati attraverso il modulo di localizzazione. Il secondo filone ha sviluppato un sistema di segmentazione del manto stradale utilizzando dati LiDAR, distinguendo aree percorribili e ostacoli. La segmentazione è stata integrata nel local re-planning per correggere dinamicamente le traiettorie. Le sperimentazioni presso il Centro Ricerche Casaccia hanno validato l'efficacia del sistema.

7 Dettaglio delle attività svolte

7.1 Introduzione al sistema di navigazione finale

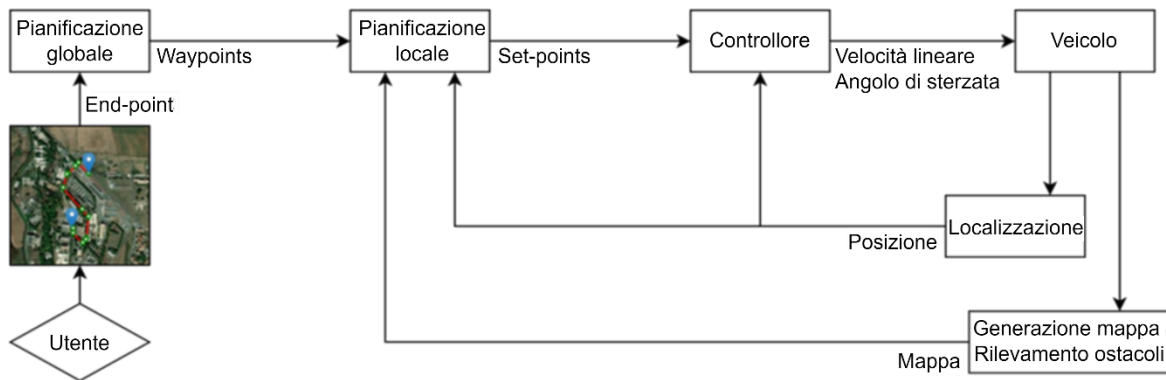


Figura 1: sistema di navigazione: il punto di arrivo viene selezionato tramite interfaccia grafica e, successivamente, il pianificatore globale GraphHopper genera una sequenza di waypoints fra la posizione attuale del veicolo ed il punto di arrivo. Il modulo di pianificazione locale calcola un percorso che congiunge gli waypoints restituiti dal pianificatore globale ed evita al contempo gli ostacoli. Il controllore in retroazione garantisce l'inseguimento del percorso fornito dal pianificatore locale.

I moduli funzionali individuati nella LA2.21 sono stati implementati come segue:

- **Localizzazione:** il modulo sviluppato comprende una fusione sensoriale per ridurre l'incertezza sulla posa fondendo informazione proveniente da un sensore GPS-RTK e una stima ottenuta da DLO, un algoritmo di odometria basata su LiDAR.
- **Pianificazione globale:** il pianificatore globale impiegato, denominato GraphHopper, permette di calcolare un percorso descritto da una sequenza di waypoints georeferenziati. A tal riguardo, è stata sviluppata un'interfaccia grafica che permette all'utente di selezionare il punto di arrivo.
- **Pianificazione locale:** una versione modificata di RRT* si occupa di calcolare un percorso per raggiungere la destinazione desiderata evitando collisioni con ostacoli, a partire dalla sequenza di waypoints forniti dal pianificatore globale e la mappa dell'ambiente circostante.
- **Generazione della mappa:** l'ambiente viene rappresentato attraverso una *occupancy grid*, o griglia di occupazione, dove ogni cella codifica non solo un'informazione di spazio libero o occupato, ma vengono distinte anche le regioni dello spazio appartenenti al manto stradale o al marciapiede, per identificare quali sono le celle percorribili dal veicolo.

- **Controllore:** per l'inseguimento del percorso è necessario implementare un sistema di controllo in retroazione che valuti in tempo reale la differenza tra la posa attuale del veicolo e la posa desiderata lungo il percorso pianificato. Questa deviazione, nota come errore di inseguimento, deve essere azzerata dal controllore in retroazione al fine di portare il veicolo sul percorso desiderato e di attenuare i disturbi che agiscono sulla dinamica dello stesso.

7.2 Modulo di Localizzazione

La localizzazione è una componente fondamentale in un sistema di navigazione autonoma, permettendo al robot di determinare la sua posizione e orientamento. Può essere fornita da infrastrutture esterne (GPS, GNSS, BLE, UWB) o stimata dai dati raccolti da sensori (LiDAR, telecamere, IMU).

In particolare, il veicolo autonomo usato nella campagna sperimentale dispone di GPS-RTK e LiDAR. Il GPS-RTK fornisce dati più accurati grazie a correzioni in tempo reale, riducendo gli errori tipici dei GPS tradizionali. I GPS-RTK richiedono due ricevitori: una stazione di riferimento in una posizione fissa e nota che riceve i segnali satellitari e calcola le correzioni, e un ricevitore mobile che riceve i segnali satellitari e le correzioni da applicare a questi dalla stazione di riferimento.

Inoltre, le nuvole di punti raccolte dal sensore LiDAR possono essere elaborate per stimare il movimento del veicolo sfruttando un processo di allineamento tra scansioni successive. DLO¹ (Direct Lidar Odometry) è l'algoritmo utilizzato per stimare l'odometria del veicolo e la mappa dell'ambiente circostante a partire dalle scansioni LiDAR.

Infine, a causa dei possibili errori delle diverse fonti odometriche, è stato deciso di fondere i dati sensoriali tramite Extended Kalman Filter (EKF) per ottenere lo stato 6DoF del veicolo (posa e velocità) utilizzando il pacchetto ROS `robot_localization`².

¹ Chen, K. et al. (2022). "Direct LiDAR Odometry: Fast Localization With Dense Point Clouds", IEEE Robotics and Automation Letters (RA-L), 7, 2.

² https://github.com/cra-ros-pkg/robot_localization

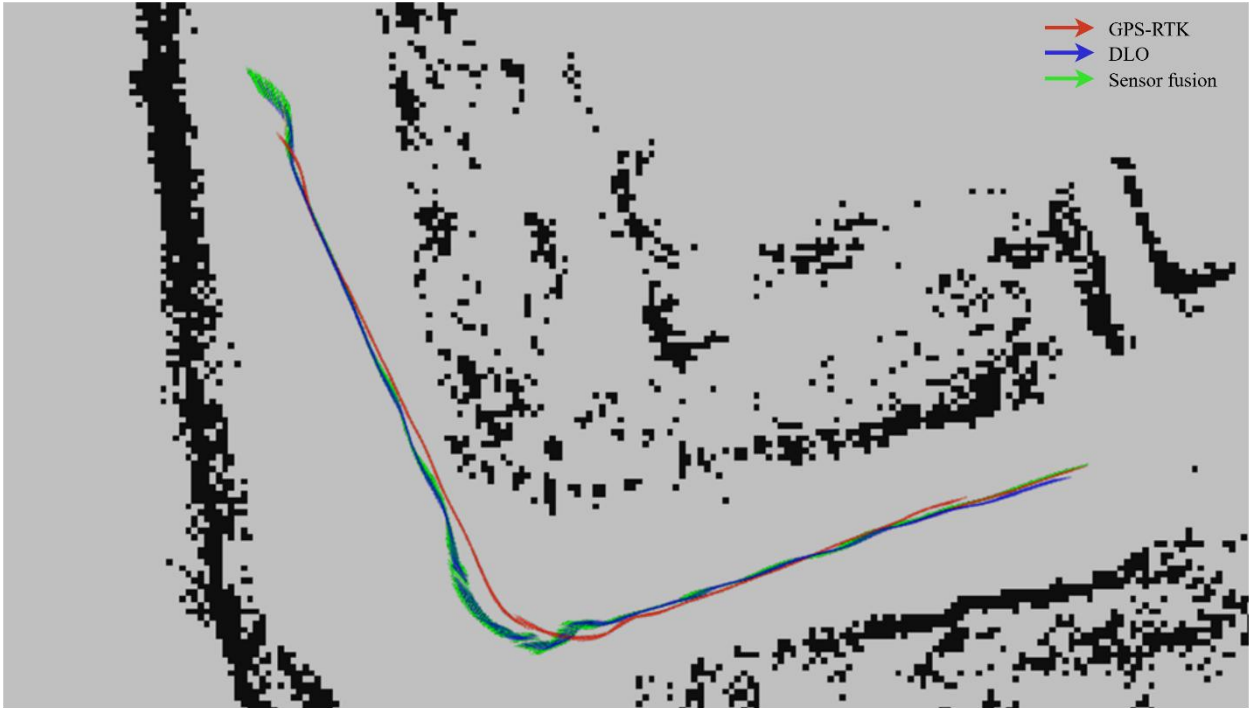


Figura 2: stima della posa attraverso GPS-RTK (in rosso), DLO (in blu), e fusione sensoriale (in verde).

7.3 Modulo Pianificazione globale

La pianificazione dei percorsi su strada basata su GraphHopper, un sistema open-source per il routing e la navigazione basato su grafi, è stata oggetto della LA2.21. È stata sviluppata un'interfaccia software che permette di gestire le coordinate GPS della posizione attuale del robot (indicata come *startpoint*) e della destinazione (indicata come *endpoint*), selezionata dall'utente sulla mappa. La mappa mostra un'immagine satellitare, la posizione corrente del robot, aggiornata ogni tre secondi, e il percorso calcolato verso la destinazione, costituito da una sequenza di waypoints.

Come anticipato nei risultati della LA2.21, le coordinate GPS degli waypoint devono essere convertite in coordinate cartesiane per essere processate dal modulo di pianificazione del sistema di navigazione. A differenza della convenzione precedentemente proposta, è stata implementata la proiezione di Gauss-Krüger, utilizzata anche dal driver Duro GPS³, per la conversione delle coordinate GPS degli waypoints in coordinate cartesiane. Le formule di mappatura per la proiezione appena citata derivano da espansioni in serie e sono date da:

$$x = N \cos(\varphi) l + \frac{1}{6} N \cos^3(\varphi) (1 - t^2 + \eta^2) l^3 + \frac{1}{120} N \cos^5(\varphi) (5 - 18t^2 + t^4) l^5 + \dots$$

$$y = B(\varphi) + \frac{1}{2} N \cos^2(\varphi) t l^2 + \frac{1}{24} N \cos^4(\varphi) t (5 - t^2 + 9\eta^2) l^4 + \dots$$

Dove $B(\varphi)$ rappresenta la lunghezza dell'arco del meridiano dall'equatore, N il raggio di curvatura nel primo verticale, λ_0 la longitudine del meridiano centrale, $l = \lambda - \lambda_0$ la differenza di longitudine, mentre t e η sono due quantità ausiliarie. Maggiori dettagli possono essere

³ https://github.com/szenergy/duro_gps_driver

trovati nel libro “Global Positioning System Theory and Practice” di B Hofmann-Wellenhof (1994)⁴.

L’output di questo modulo è quindi costituito dalla sequenza di waypoints convertita in coordinate cartesiane relative rispetto alla posizione iniziale del veicolo.

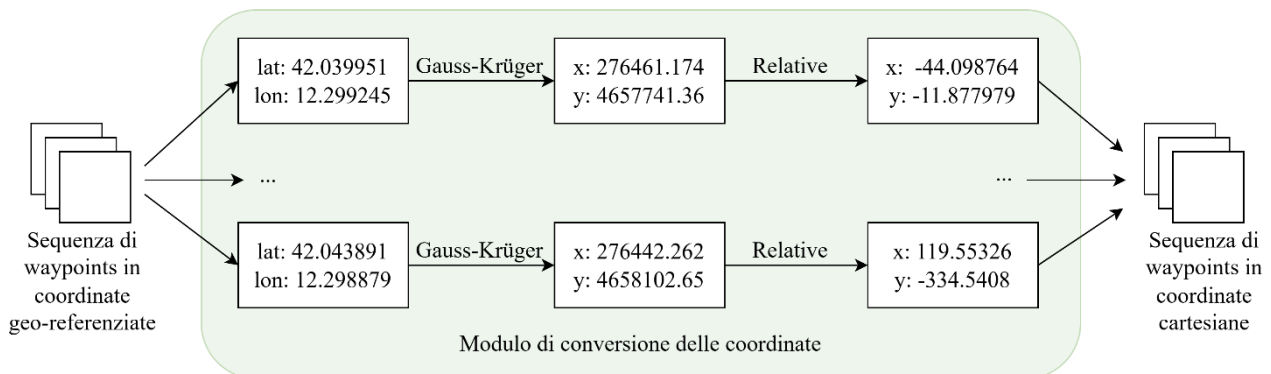


Figura 3: il modulo di conversione delle coordinate è una componente fondamentale del pianificatore globale in quanto, a partire dalla sequenza di waypoints in coordinate geo-referenziate, si occupa di convertirle prima in un sistema di riferimento cartesiano attraverso la proiezione di Gauss-Krüger, per poi convertirle nel sistema di riferimento locale relativo alla posizione iniziale del veicolo.

7.4 Modulo di Pianificazione Locale

La pianificazione locale del percorso per arrivare dalla posizione attuale al prossimo waypoint individuato dal modulo di pianificazione globale si basa su una versione modificata dell’algoritmo RRT* (Rapidly-exploring Random Tree Star) in cui vengono utilizzate le primitive di Dubins. L’algoritmo RRT* rappresenta un’evoluzione del classico RRT, in cui viene pianificato un percorso costruendo iterativamente un albero all’interno dello spazio libero per evitare collisioni con gli ostacoli. In modo simile a RRT, l’albero viene generato campionando in maniera randomica nodi nello spazio libero e, per ogni nuovo nodo, viene controllato che esso e il relativo percorso che lo connette non entrino in collisione con gli ostacoli. A differenza di RRT, RRT* include un processo di ricampionamento e ottimizzazione dell’albero al fine di ridurre la lunghezza e il costo totale del percorso. Infine, la versione implementata utilizza i percorsi di Dubins al fine di rendere le traiettorie compatibili con il modello cinematico del veicolo (definito nella LA2.21). In particolare, un percorso di Dubins definisce la curva più corta che connette due punti nello spazio (x, y) per un veicolo non-olonomo il cui movimento è limitato alla traslazione in avanti e rotazione vincolata da un minimo raggio di curvatura. Il percorso ottimo consiste nel percorso più breve combinando tre segmenti appartenenti alle classi R (curva a destra), L (curva a sinistra), e S (segmento rettilineo).

⁴ Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1994). “GPS: Theory and Practice”, 3rd ed. New York: Springer-Verlag Wien.

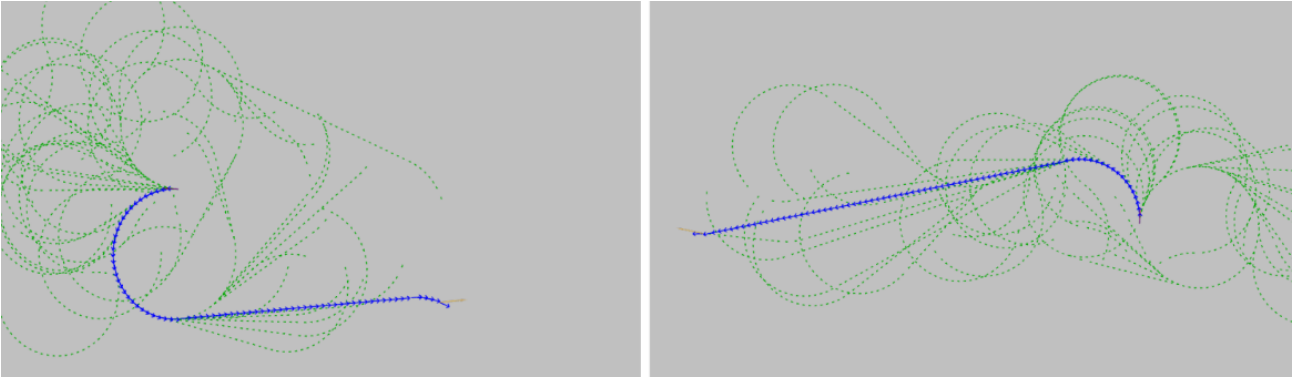


Figura 4: esempi di percorsi generati da RRT*-Dubins. L'algoritmo genera un albero di rami candidati (in verde), tra i quali seleziona il percorso con costo minore per arrivare a destinazione (in blu).

In conclusione, questo modulo restituisce un percorso p_t definito da una sequenza di set-point di posa $s_{t,i}$ per raggiungere il prossimo waypoint w_{t+1} :

$$p_t = [w_t, s_{t,1}, \dots, s_{t,i}, \dots, w_{t+1}]$$

Una volta inviato il percorso attuale p_t al controllore, il pianificatore locale si occupa parallelamente di calcolare il prossimo percorso ottimo p_{t+1} che collega l'ultimo punto w_{t+1} del percorso attuale al waypoint successivo w_{t+2} . Nel caso di un ostacolo dinamico, non visibile durante la generazione del percorso, il veicolo rileva in tempo reale la possibile collisione e si ferma per pianificare un nuovo percorso che eviti l'ostacolo.

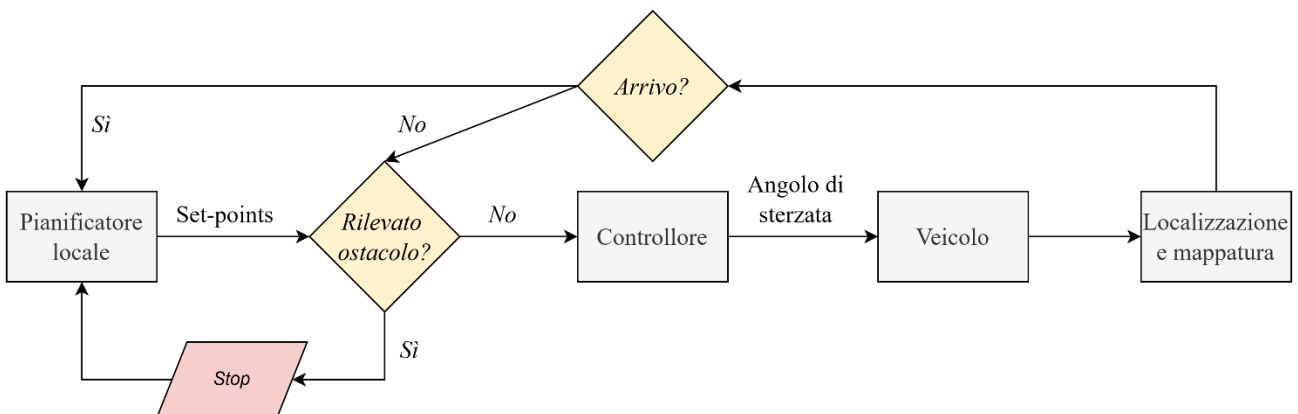


Figura 5: durante l'inseguimento del percorso generato dal pianificatore locale, questo modulo si occupa di rilevare possibili collisioni con ostacoli lungo il percorso stesso. In caso positivo, il veicolo viene fermato finché non viene ripianificato un nuovo percorso che eviti l'ostacolo identificato.

7.5 Modulo di Generazione della Mappa e Segmentazione del Manto Stradale

È fondamentale per un veicolo autonomo costruire una rappresentazione accurata dell'ambiente circostante per potersi localizzare e in seguito pianificare un percorso privo di ostacoli per giungere alla destinazione desiderata. Ai fini della navigazione autonoma, è fondamentale generare una rappresentazione detta *occupancy grid*, ovvero una rappresentazione a griglia in cui ogni cella contiene una variabile binaria che codifica la presenza o assenza di un ostacolo. In particolare, ogni cella (x_i, y_i) ha uno stato associato m_i , che può essere occupato ($m_i = 1$) o libero ($m_i = 0$).

Per la generazione della rappresentazione a griglia, è possibile elaborare la nuvola di punti restituita dal sensore LiDAR utilizzando specifici algoritmi di mappatura. Ad esempio, gli algoritmi di odometria basata su LiDAR stimano il movimento del veicolo calcolando la trasformazione necessaria per allineare nuvole di punti ottenute da scansioni consecutive tramite algoritmi di allineamento come ICP (Iterative Closest Point). Come specificato nella Sezione 7.2, DLO restituisce la stima odometrica insieme ad una nuvola di punti rappresentante la mappa 3D dell'ambiente circostante. Questi punti 3D sono proiettati nello spazio 2D per poter popolare le celle della griglia di occupazione tramite il nodo ROS `pointcloud_to_laserscan`⁵, grazie al quale possiamo anche escludere dal calcolo degli ostacoli punti troppo alti (come i rami degli alberi) impostando la variabile `~max_height`.

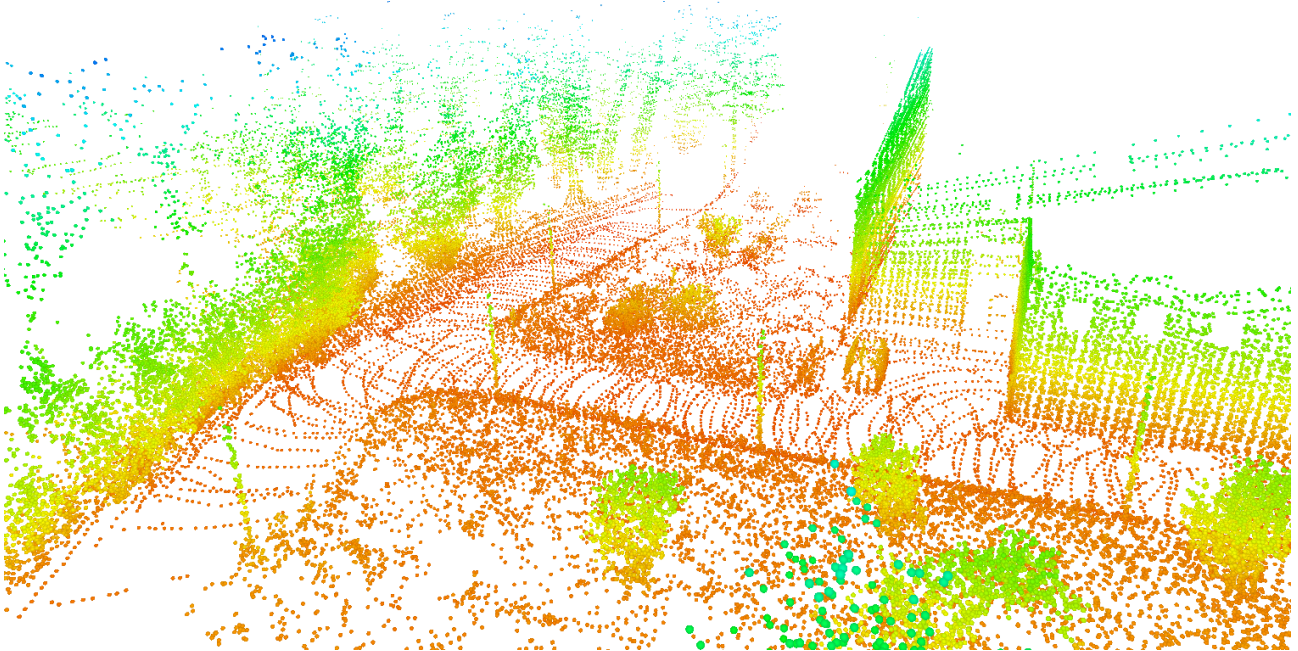


Figura 6: mappa 3D generata da DLO a partire da misure LiDAR.

Tuttavia, all'interno di un'infrastruttura stradale è necessario identificare elementi cruciali nell'ambiente, come la strada e il marciapiede. Per questo scopo, la classica rappresentazione a griglia può essere estesa ad una griglia di occupazione semantica, dove è possibile integrare le informazioni di occupazione tradizionali con dettagli semantici. In questo caso, lo stato m_i^c può rappresentare la presenza di un oggetto della classe c nell' i -esima cella. Ai fini del progetto, abbiamo scelto di elaborare la nuvola di punti restituita dal LiDAR con un algoritmo di segmentazione⁶ (la cui implementazione è disponibile pubblicamente⁷) che si occupa di determinare se ciascun punto 3D nella nuvola di punti appartiene alla strada o al marciapiede. Come nel caso precedente, i punti 3D stimati come punti appartenenti al marciapiede vengono proiettati nello spazio 2D per popolare la rappresentazione a griglia.

In conclusione, la rappresentazione dell'ambiente utilizzata è costituita da una griglia di occupazione in cui le celle libere corrispondono alle regioni dello spazio percorribili dal veicolo.

⁵ https://wiki.ros.org/pointcloud_to_laserscan

⁶ Horváth, E., Pozna, C., and Unger, M. (2022). "Real-Time LIDAR-Based Urban Road and Sidewalk Detection for Autonomous Vehicles". *Sensors*, 22, 1.

⁷ https://github.com/jkk-research/urban_road_filter

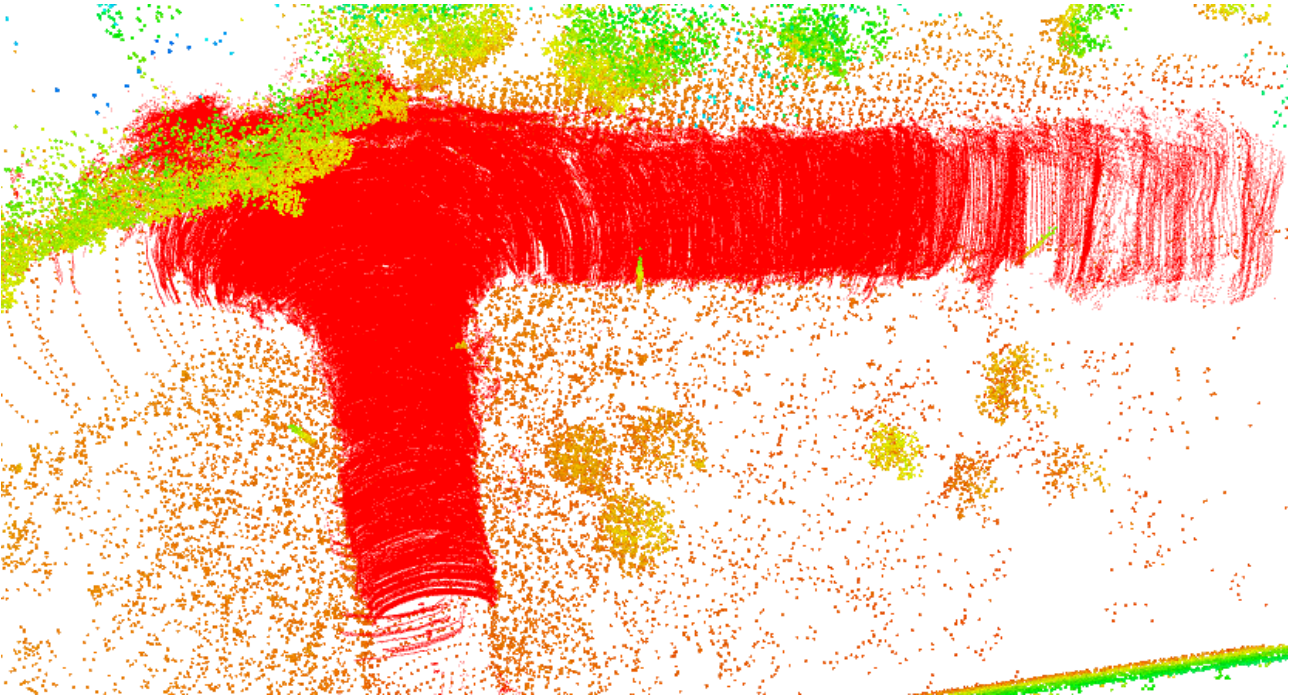


Figura 7: esempio di segmentazione del manto stradale (in rosso) a partire dalle misure LiDAR.

Per ridurre il rumore causato da rilevazioni spurie del sensore o classificazioni errate, l'aggiornamento della mappa si basa su una formulazione basata su framework Bayesiano, il quale fornisce un algoritmo per combinare informazioni ottenute da multiple osservazioni. Come primo step, i valori della griglia M_t ottenuti dalla lettura attuale del LiDAR vengono trasformati in probabilità di occupazione $p(m_i|z_t)$:

$$p(m_i|z_t) = p_L M_{i,t} + (1 - p_L)(1 - M_{i,t})$$

Dove p_L rappresenta l'affidabilità che assegniamo al sensore. Successivamente, le probabilità di occupazione vengono prima convertite in scala logaritmica:

$$l_{i,t} = \log \frac{p(m_i|z_t)}{1 - p(m_i|z_t)}$$

Le probabilità di occupazione note le prime t osservazioni sono date da:

$$l_{i,1:t} = l_{i,1:t-1} + l_{i,t} + l_0$$

Dove l_0 rappresenta la probabilità a priori di occupazione. A posteriori del processo di fusione, è possibile recuperare le probabilità di occupazione di ogni cella:

$$p(m_i|z_{1:t}) = \frac{1}{1 + \exp(-l_{i,1:t})}$$

L'aggiornamento della mappa appena descritto viene però applicato solo ad una mappa locale intorno al veicolo, mentre le celle di occupazione nella mappa globale non appartenenti alla mappa locale vengono mantenute invariate rispetto all'ultimo valore individuato nel corrispondente processo di aggiornamento.

7.6 Controllore

Il sistema di controllo impiegato per l'inseguimento del percorso, definito da una sequenza di set-points di posa, si basa sul metodo di Stanley⁸, oggetto della LA2.21. Il controllore agisce sull'angolo di sterzata della vettura $\delta(t)$ per compensare l'errore di posizione (e) e l'errore di orientamento (ψ) come segue:

$$\delta(t) = \psi(t) + \tan^{-1} \left(\frac{ke(t)}{k_s + v(t)} \right), \quad \delta(t) \in [\delta_{min}, \delta_{max}]$$

dove $[\delta_{min}, \delta_{max}]$ è l'intervallo che contiene i valori ammissibili dell'angolo di sterzata. In particolare, è necessario innanzitutto individuare il set-point di posa più vicino al centro geometrico dell'asse anteriore del veicolo, e in seguito calcolare le due componenti del segnale di controllo che correggono gli errori di orientamento (ψ) e di posizione (e).

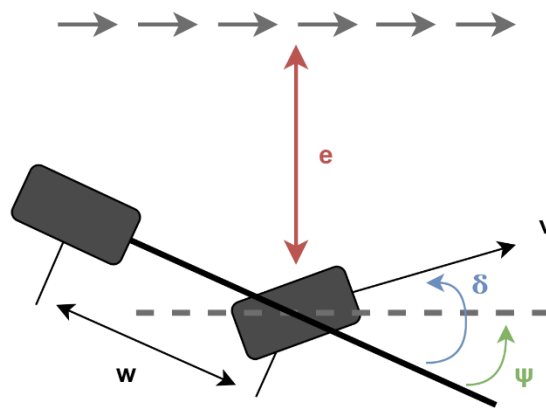


Figura 8: il controllore di Stanley agisce congiuntamente l'errore di traiettoria (ψ) e l'errore di posizione (e) per l'inseguimento del percorso.

7.7 Integrazione dei moduli e ROS

Come indicato nella LA2.21, la gestione della comunicazione fra i diversi moduli di navigazione è affidata a sistema operativo ROS (Robot Operating System). Tutti i nodi ROS sviluppati comunicano scambiando messaggi su dei topic, favorendo la scalabilità e la flessibilità del sistema. L'unica eccezione riguarda la serie di waypoints generati dal pianificatore globale, per cui è stato implementato un servizio ROS ad-hoc al fine di garantire uno scambio di messaggi sincrono.

⁸ Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., et al. (2007). "Stanley: The Robot That Won the DARPA Grand Challenge". The 2005 DARPA Grand Challenge. Springer Tracts in Advanced Robotics, 36, 1.



Figura 10: veicolo autonomo utilizzato nella campagna sperimentale.

7.9 Risultati ottenuti

Per validare l'architettura di navigazione autonoma proposta, è stata svolta una campagna sperimentale all'interno del Centro Ricerche Casaccia. In particolare, è stata sia verificata la corretta funzionalità delle singole componenti, come il modulo di localizzazione, di pianificazione, o controllo, sia l'architettura complessiva.

Gli esperimenti hanno previsto l'esecuzione di traiettorie sia rettilinee che curvilinee, al fine di valutare le prestazioni del sistema in due scenari distinti: quando il manto stradale è già visibile e quando la mappa dello spazio libero viene aggiornata dinamicamente in seguito a una curva.



Figura 11: veicolo utilizzato durante una sperimentazione di guida autonoma.

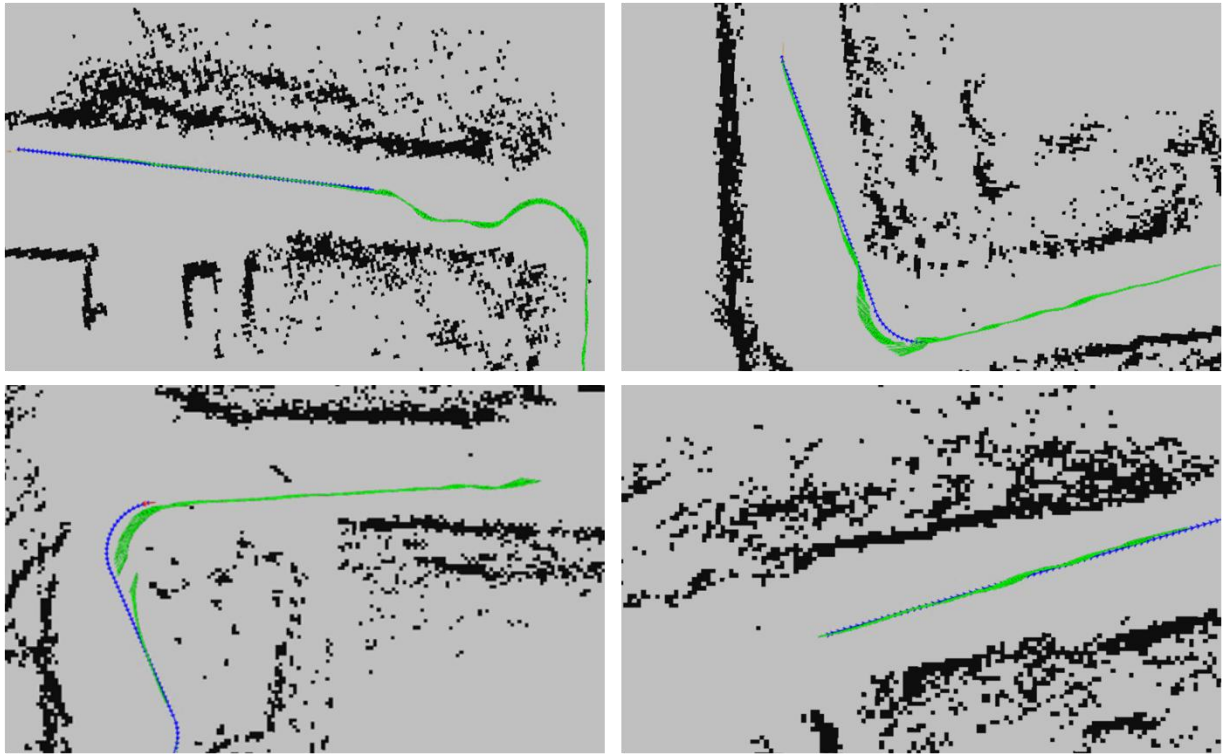


Figura 12: esempi di navigazione autonoma all'interno del Centro Ricerche Casaccia. Il controllore si occupa di inviare i comandi ai motori per annullare l'errore di inseguimento tra il percorso desiderato (in blu) e la posa del veicolo (in verde).

8 Contributo delle eventuali consulenze alle attività sopra descritte

Nello sviluppo delle varie attività previste per questa LA non è stato fatto ricorso a consulenze esterne. Tutte le attività qui descritte sono state sviluppate dal Team di ricercatori UniPg.

9 Pubblicazioni scientifiche

Elenco delle pubblicazioni scientifiche eventualmente risultanti dall'attività svolta.

- S. Felicioni, E. Burani, M. Leomanni, M. L. Fravolini, P. Valigi and G. Costante, "Integrating Occupancy Grid with Semantic Road Information for Autonomous Navigation in Urban Scenarios: A Benchmark Study," 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 2024, pp. 2665-2671, doi: 10.1109/CASE59546.2024.10711541.

10 Eventi di disseminazione

Dalla presente Linea di Attività non sono scaturiti eventi di disseminazione.